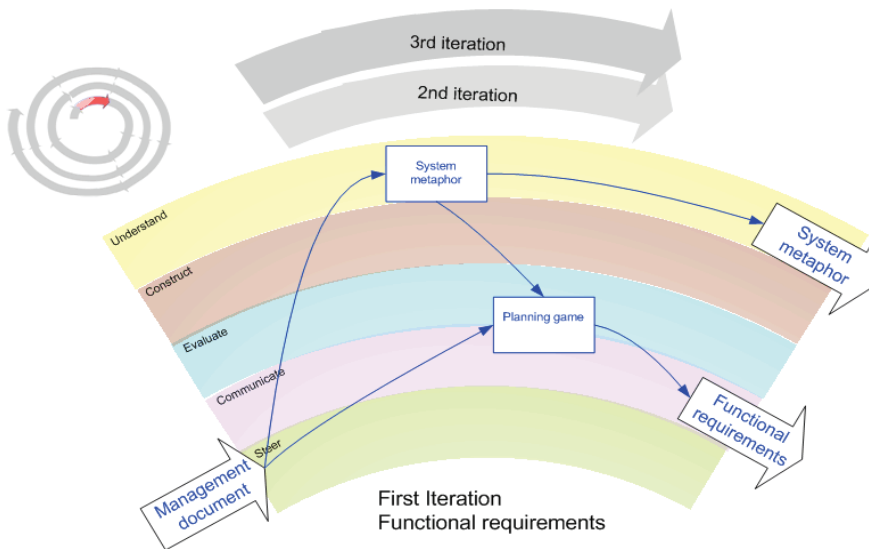


## ITERATION 1: FUNCTIONAL REQUIREMENTS

### SYSTEM METAPHOR



Bite: Describe the system by reference to some other thing or concept

Time: 2 hours

Input: Notes from first meeting with client

Process evidence: Annotated notes

Client: Discussion

### WHAT IS A METAPHOR?

In the first iteration, the functional requirements sector involves us having discussions about our approach to the business problem or opportunity. We do this by means of a system metaphor.

A metaphor is a comparison between two seemingly unrelated subjects. They are used in language to enliven, to encourage interpretation and to provide a vehicle for understanding when either there are no direct terms for a concept or other explanation is cumbersome. By understanding and experiencing one thing in terms of another we can provide a means of exploring a concept even before we've really come to terms with what it is we're talking about.

Perhaps the most famous metaphor is Shakespeare's opening:

*All the world's a stage,  
And all the men and women merely players  
They have their exits and their entrances;*

(William Shakespeare, As You Like It, 2/7)

In ordinary conversation we speak of "pulling your socks up", "drowning in paper", "unfolding the road map to peace" and "my memory is a little foggy". The metaphor we use for something can be quite informative. For example, the words we use for business are often based on a war metaphor: "hostile takeover", "losing ground", and so on, and these in turn affect the management of the business. Ray Jackson (quoted in CEO Forum) uses lessons from the military in a discussion of business practices:

*I think it taught me that the world is always imperfect, and that your knowledge base is also imperfect. In military operations, you never really have 'all the facts', in the literal sense. So it teaches you to operate on incomplete data: take the data you have, turn it into information, and figure out what you need to do. I guess I learnt that clarity is better than certainty: there comes a time when you have enough information to move on. If you wait for certainty it never arrives. The military makes you comfortable with uncertainty.*

*The other thing I learnt was the need to not procrastinate in making decisions, and to be incisive. You learn a process of continually making decisions, then reviewing those decisions as more information comes to light. This has been very useful in business.*

*The military also teaches you to carefully manage resources. On a naval mission, for instance, you only have a limited amount of fuel, a limited amount of ammunition, and so on. Your most important limited resource, of course, is your people, so you need to look after and nurture them - you can't work them to death! The military teaches you to always be very mindful of your resources and how you can best use and conserve them.*

*I also think you learn a lot about leadership and communication. I like the expression 'People don't want to be continually inspired, they just want to know what they should do'.*

*Most military people have a highly competitive, 'we are here to win' streak, so that toughness and commitment is there. Some aspects of business are like a battle, so those types of attitudes can be useful.*

*Another useful attitude is to set your goals, and then be steadfast in pursuing them come what may. The goals of war and business are different, of course. War is about defeating an enemy - it's a zero sum game. In business, on the other hand, your objective may be to be the most profitable company. It's not a goal that involves defeating an enemy in the same way - it's more about achieving your own goals.*

(See also Marketing's maneuver theory for explicit - and controversial use of the military/business metaphor: "The Armory - the online resource for the competitive marketer is the exclusive host for The Art of Attack - How to Attack and Dislodge the Larger Competitor").

There are similar metaphors such as "time is money", "women are from Venus" etc, indeed, Lakoff and Johnson (1980) argue that we live our life by metaphors.

## SOFTWARE DEVELOPMENT: SYSTEM METAPHOR

In software development the System Metaphor has been adopted as a core practice by the agile community. Kent Beck, author of Extreme Programming Explained defines a system metaphor as:

*"a story that everyone - customers, programmers, and managers - can tell about how the system works." p. 179.*

Wake argues that we seek a system metaphor for several reasons:

**Common Vision:** To enable everyone to agree on how the system works. The metaphor suggests the key structure of how the problem and the solution are perceived. This can make it easier to understand what the system is, as well as what it could be.

**Shared Vocabulary:** The metaphor helps suggest a common system of names for objects and the relationships between them. This can become a jargon in the best sense: a powerful, specialized, shorthand vocabulary for experts. Naming something helps give you power over it.

**Generativity:** The analogies of a metaphor can suggest new ideas about the system (both problems and solutions). For example, the metaphor, "Customer service is an assembly line". This suggests the idea of a problem being handed from group to group to be worked on, but it also raises the question, "What happens when the problem gets to the end of the line - does it just fall off?" This can bring out important issues that might otherwise lurk and fester.

**Architecture:** The metaphor shapes the system, by identifying key objects and suggesting aspects of their interfaces. It supports the static and dynamic object models of the system.

Choosing a metaphor takes work. Hopefully, you'll traverse deep and rich ground as you come to agreement

over an appropriate metaphor. This is an important part of the process and you should attempt to recognise and capture these thoughts. Jefferies (2001), in an evocative description of an agent-based information retrieval system describes “this program works like a hive of bees, going out for pollen and bringing it back to the hive”. Although Jefferies doesn’t describe them, it is likely that they tried other metaphors first. Perhaps “a vacuum cleaner sucking up all knowledge?...no there’s lots of them and anyway the information is structured - not just a big dustball, well perhaps a library? no, that is big and static, what we want is something that selectively collects information, like a fleet of recycling trucks...”.

Some metaphors are used repeatedly in software development. Shaw and Garlan identify common approaches:

1. Spreadsheet Metaphor
- 2 Script Metaphor
- 3 Manufacturing Metaphor (e.g. LinesStationsBinsParts or AssemblyLine)
- 4 Accounting Metaphor (double-entry archive notation)
- 5 Shopping Cart Metaphor (e-commerce)
- 6 Auction Metaphor (e-commerce)
- 7 Blackboard Metaphor (ai)
- 8 Document Processor (desktop systems where the “model” gets saved as a file)
- 9 Virtual Space Metaphor (eg. VR)
- 10 Desktop Metaphor
- 11 Tools and Materials Metaphor
- 12 Buttons Everywhere Metaphor

In cases where no one can think of an appropriate metaphor (with or without vivid imagery), the approach is to develop a “naive metaphor”. This means that you describe the system more literally (a student management system would have student and enrolment objects etc). The disadvantage of this is that it isn’t really a metaphor - you won’t gain any deeper understanding or insights as it is rooted in what you already know.

## PROCESS

So, what do we do with this system metaphor? For some Agile proponents, the system metaphor forms that basis for much of the programming, the tenor and subjects of the metaphor directly translating to classes and patterns.

We take a lighter approach. The system metaphor is primarily a tool for discussion here in the functional requirements of the first iteration. It may also prove useful to prompt thought in the Design Concepts of Iteration 2.

The metaphor should be stated in simple terms. As Ryan states: “The system is a bakery” jibes better than “The system interprets text as commands and executes them against builders that produce resultant objects and attach assorted decorators, sorting them via pipes and filters to the correct bins; the user can then browse and eat them as needed.” (C2 2006).

Once you have a metaphor, the trick is to explore all of its characteristics, writing them down on paper or whiteboard. These ideas will flow directly to the user stories or functional requirements.

For a project to develop a project management and group collaboration tool, a metaphor of a fridge door was useful.

For me the project is a fridge door:

- always there (you don't have to open it to find information),
- messages to family (from each other, phone messages)
- calendar (today, arrangements, upcoming events)
- shopping list
- security strap (to stop little chocolate stealers)
- inspiration ("do you need that chocolate?" sticker)
- progress charts (weight gain/loss)
- family cleaning/cooking roster
- current work (especially for younger ones, sketches named and sometimes dated)
- document repository (music tickets, bills to pay) - interestingly layered!
- reconfigurable (information held together by fridge magnets)
- teaching role (alphabet letters introducing words and information to little ones).
- star chart (kids' behaviour management, behaviour contract and rewards).
- public place,
- central place: we go past it as part of normal life (work)flow (but without it jumping out and interrupting)
- achievements (kids' school certificates)
- photos of family (selves and relatives/baby photos)
- newspaper clippings
- limited space, information has to be managed to avoid loss in clutter (but this is done without any rules, design or manager) (note, I did write "it does this" here, the door itself of course doesn't do anything except act as a static repository, it is the family who actively manage the information).
- phone numbers (family/friends/doctor's etc)
- menu/phone number for pizza place
- ...and does all these things without getting in way of real job (keeping food cold - our system mustn't get in way of the project being undertaken).

An alternative metaphor for this project was a car dashboard, another was a library.

For another project, a community based surf report system for surfers, a single system metaphor proved elusive. The process, however, was very useful. As we explored possible system metaphors - a newspaper; a radio snow report; a school newsletter - the reasons why these each of these didn't suit helped identify the characteristics of the desired system.

## Dangers of metaphors

While a different metaphor might be better, it is unlikely that a metaphor can be wrong, especially if you are using it as a tool for exploration. If you realise that another metaphor is better, then good, it means that you have made a breakthrough in your understanding.

Rather than "wrong" the worst outcome is that the metaphor becomes a cause of miscommunication. The following is extracted from the [c2.com/xp wiki](http://c2.com/xp/wiki/):

*Ask yourself, what more familiar things is this problem like? Is it really like ordering coffee from a fancy coffee machine? Is it really mostly like steering (tacking) a sailboat across a lake? driving from Toronto to Paris? (...a journey of less than 90 minutes, since Paris is just on the other side of Brantford. -- RandyMacDonald) (just to avoid the CanadianCulturalAssumption)*

*You mean Paris, Kentucky, right? :) Or perhaps a polar route during a cold winter...*

*\* Whoever tried to clean this up by deleting didn't fix the underlying problem, so I put it back. Next WikiGnome: clarify the driving-from-Toronto-to-Paris issue or leave as is for now.*

*No, best left! It illustrates rather beautifully the way subtle misunderstandings can arise. Developer-One says: "This is like driving to Paris", meaning a short journey. Developer-Two gets a mental image of driving through the Chunnel from the UK to France, and considers how to avoid trains. Developer-Three (in the USA) ponders submarine cars. The misunderstandings here highlight the importance of MeaningDependsOnContext in discussions, especially about something as crucial as the SystemMetaphor. -- BenLast*

We have to be careful then , to avoid "false insights".

A metaphor needs to be familiar enough to the group that needs to use it. To say that it is "an accrual account system but for carbon credits" might be a perfect metaphor for the financial accountants but if such accounting isn't understood by the programmers, then it will act to confuse rather than enlighten.

Sometimes the metaphor limits understanding. The paper based rows and columns spreadsheet was a strong model in the development of spreadsheets (!), but the conceptual leap from the paper to the real power of a modern spreadsheet is quite large. This leads to the danger of introducing "magic": "it's like a magic whiteboard". While this helpfully encompasses all of the basic functions of a whiteboard, the real innovation of the system is hidden behind a veneer of "magic".

We also need to know when to give up on the metaphor. The project management development described above by the fridge door metaphor was also described by a car dashboard. This worked extremely well as a system metaphor in early stages: unobtrusive but critical monitoring of vital indicators etc, but became cumbersome when used as the basis for the interface design (dials, steering wheel with leather trim and so on).

## REFERENCES

<http://en.wikipedia.org/wiki/Metaphor>

<http://c2.com/xp/SystemMetaphor.html> last updated 15/2/2006 Viewed 24/7/6

Jackman, R. (2006) Is business war? What military affairs can (and can't) teach CEOs [http://www.ceoforum.com.au/200206\\_ceoinsight.cfm](http://www.ceoforum.com.au/200206_ceoinsight.cfm) Viewed 21/7/6

Jefferies, R. (2001) What is eXtreme Programming <http://www.xprogramming.com/xpmag/whatisxp.htm> viewed 21/7/6

Lakoff G. & Johnson, M. (1980) Metaphors We Live By <http://theliterarylink.com/metaphors.html>

Wake, W. (2000) System Metaphor <http://www.xp123.com/xplor/xp0004/index.shtml> viewed 21/7/6

Wake, W. (2000) Proven system metaphors <http://c2.com/cgi/wiki?ProvenSystemMetaphors> viewed 21/7/6  
<http://knowgramming.com/>

[http://twoscenarios.typepad.com/maneuver\\_marketing\\_commun/2005/05/military\\_metaph.html](http://twoscenarios.typepad.com/maneuver_marketing_commun/2005/05/military_metaph.html)